(19) Europäisches Patentamt
European Patent Office
Office européen des br v ts

(11) EP 0 867 814 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
30.09.1998 Bulletin 1998/40

(51) Int Cl.[6]: **G06F 13/364**

(21) Application number: **98302080.1**

(22) Date of filing: **19.03.1998**

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **25.03.1997 US 823736**

(71) Applicant: **International Business Machines Corporation**
Armonk, N.Y. 10504 (US)

(72) Inventors:
- **Revilla, Juan G.**
  Cary, North Carolina 27513 (US)
- **Satorius, Thomas A.**
  Raleigh, North Carolina 27614 (US)
- **Schaffer, Mark M.**
  Cary, North Carolina 27511 (US)

(74) Representative: **Davies, Simon Robert**
IBM,
United Kingdom Limited,
Intellectual Property Law,
Hursley Park
Winchester, Hampshire SO21 2JN (GB)

(54) System and method for controlling a bus

(57) Bus performance in a computer system having multiple devices 102, 104, 106 accessing a common shared bus 132 may be improved by increasing throughput and decreasing latency while accounting for dynamic changes in bus usage. Devices submit a priority level along with a bus request to a bus controller 120. Upon receiving multiple requests, an arbiter of the bus controller 130 compares the priority levels associated with the different bus requests and grants control of the bus to the device having the highest priority level. During each cycle that a device has control of the bus, a feedback logic circuit of the bus controller determines whether other bus requests are pending, and if so, determines the highest pending request priority level. Signals corresponding to the results of these determinations are fed back to each device. The device having control of the bus uses the combination of the currently pending request priority level and the device's own latency timer 102a, 104a, 106a to determine whether it should maintain control of the bus or relinquish control of the bus. If the latency timer of the device has not expired, the device will continue to control the bus even if the currently pending request priority level is greater than the device's priority level. If the currently pending request priority level is not greater than the device's own priority level, the device will continue to control the bus even after the device's latency timer has expired until the device no longer needs the bus. Finally, if the currently pending request priority level is greater than the device's own priority level and the device's latency timer has expired, the device will relinquish control of the bus. As such, the latency timer of a device is dynamically extended after expiration as long as the device has a priority level at least as great as the currently pending request priority.

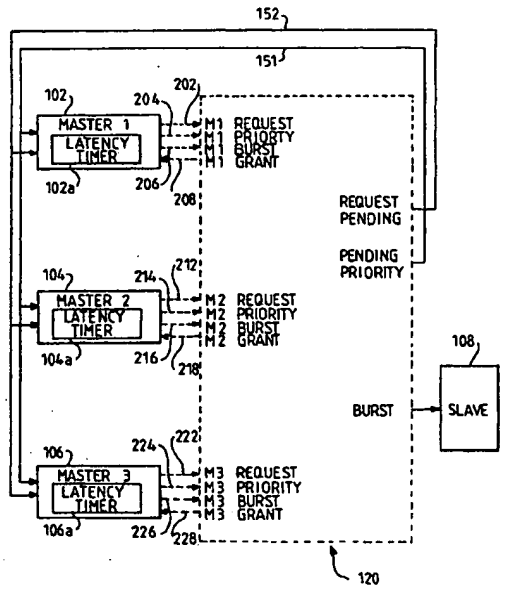

FIG. 2

EP 0 867 814 A2

Description

The present invention relates to bus performance, and more particularly, to dynamically controlling a bus having a plurality of devices operationally connectable thereto to improve throughput and decrease latency.

The number of functions on a single integrated circuit chip continues to increase in concert with an increase in chip densities. These "system-on-a-chip" integrated circuits typically use a common, shared bus architecture to provide the communication link between the various devices and subsystems of the "computer system." A common bus provides a low cost communication link since it can be shared between multiple devices in the computer system. However, the linking of multiple devices to a single bus may raise concerns over maximum bus performance.

Achieving maximum bus performance may be difficult in a shared bus architecture. Factors which severely impact bus performance include system throughput (i. e., bandwidth) and system response time (i.e., latency). For purposes of determining throughput or bandwidth, a bus transaction is a bus transaction completed by the device which is on the receiving end of the transmission. Throughput or bandwidth is the average number of bus transactions over a period of time. Response time or latency is the time it takes to complete a bus transaction for a particular device beginning with the cycle during which the device first requests the bus until the cycle that the last piece of data is transferred from the device across the bus to a second device. A device which requests access to or control of a bus and transmits and receives data across a bus may be referred to as a "master". A device which transmits or receives data across a bus and is responsive to a master may be referred to as a "slave". A slave cannot request access to or control of a bus.

In order to achieve a high degree of bus performance, the throughput must be high while the latency must be low. Further, in order to achieve a high level of bus throughput, the slave preferably is never idle and, consequently, the bus is preferably never idle. In contrast, however, since latency refers to the time it takes to complete a bus transaction beginning with the cycle during which a master first requests the bus until the cycle during which the last data is transferred by the master across the bus, latency includes the time during which a master waits for the bus to become available (i. e., idle). As a result, latency is reduced by allowing the bus to be idle.

A number of different architectural designs have been developed in an effort to address bus performance, including throughput and response time. Many of these schemes rely on the "priority" level of the device seeking control of the bus. For example, United States Patent No. 5,438,666 to Craft et al. describes an arbitration system for controlling access to a bus. The arbitration system of Craft et al. interrupts the control of the bus by a first device when a second device having greater priority requests access to the bus. Once the second device completes its access to the bus, control of the bus is returned to the first device. The transfer of control is implemented without requiring the timing overhead of arbitrating priority between bus masters having active bus requests.

United States Patent No. 5,140,680 to Best describes a bus arbitration system for a computer network having multiple master and slave devices which share a common bus. The bus arbitration system includes bus arbitration logic in each master device, and accounts for the slowest master's operational delay when determining which master shall have access to the bus at a given time.

By way of further example, United States Patent No. 5,388,228 to Heath et al. describes an arbitration system having a central arbitration control circuit and a local arbiter associated with each device seeking access to a common bus. Heath et al. also provides for the programming of each device to operate in either a linear mode or a fairness mode. When operating in the fairness mode, a first device having access to the bus in response to a second device requesting bus access will relinquish control of the bus, once the first device has completed an appropriate number of transfers, allowing the requesting device having the next highest priority level to gain control of the bus.

Other designs which involve sharing of a common bus have attempted to address the conflicting design requirements of high throughput and low latency by using long burst transfers to achieve higher throughputs while using master latency timers to reduce latency by limiting the length of the bursts. Latency timers typically may be implemented in a master using a programmable register and a counter. The initial latency count value which represents the maximum number of clock cycles that the master may maintain control or ownership over the common bus is loaded into the programmable register. The counter is typically reset to zero each time the device gains control of the bus. Once the value of the counter reaches the value stored in the register (i.e., the latency timer has expired), the corresponding device having control over the bus must relinquish the control regardless of the system's bus usage conditions.

As a result, in systems where the bus usage is light (i.e., data transfer across the bus is relatively minimal), a device whose latency timer has expired, and consequently, must relinquish control of the bus even though it has additional data to transfer across the bus, has a bandwidth (i.e., throughput) which may be unnecessarily limited. Moreover, in systems where bus usage is relatively high, in that various devices are requesting the bus simultaneously, a device is likely to use the bus until its latency timer expires. One device's control of the bus until its latency timer expires causes other devices to wait for the bus to become available, and consequently, experience relatively high latencies.

Moreover, the bus usage conditions in a system may vary over time from, for example, light usage to heavy usage back to light usage. Consequently, it may be necessary to update each device's latency timer in order to achieve maximum bus performance. A latency time may be updated by reprogramming the register and counter. However, significant overhead is required to reprogram each device's latency timer, particularly if it occurs on a regular basis. Thus, use of latency timers to improve bus performance is generally ineffective due to dynamically changing bus usage conditions.

Although various arbitration schemes or latency timers are presently used to control access to a common bus in multiple device systems, these prior designs may not effectively address the conflicting design requirements of high throughput and low latency. Moreover, these prior designs do not account for dynamically changing bus usage conditions. In order to improve maximum bus performance, the conflicting problems of high throughput and low latency must be addressed while accounting for dynamically changing bus usage conditions.

Accordingly, the invention provides a method for dynamically controlling access to a bus having a plurality of devices operationally connectable thereto, each of said devices having a priority level associated therewith, said method comprising the step of:

controlling the duration of control of the bus by a first of said devices based on the combination of a latency timer and the priority level associated with the first device, and a priority level associated with a pending request by a second of said devices to control the bus.

In a preferred embodiment, said controlling step includes the steps of: granting by a bus controller a request from the first device to control the bus; receiving at the bus controller a request to control the bus and a priority level associated with the request from the second device; and transmitting from the bus controller the priority level associated with the request received from the second device and an associated pending request signal to the first device in response to the receipt of the request from the second device. Thus the first device first receives control of the bus from the bus controller in response to said granted request; and then receives from the bus controller the priority level associated with the pending control request from the second device. The priority level of the first device is then compared with the priority level associated with the request from the second device, resulting in an extension of the duration of control of the bus by the first device if the priority level associated with the first device is at least as great as the priority level associated with the second device, or termination of the control of the bus by the first device if the priority level associated with the first device is less than the priority level associated with the second device. Note that such extension or termination of control (and indeed optionally said comparison) occur only after the latency timer associated with the first device has already expired; otherwise there is continued control of the bus by the first device until the latency timer associated with the first device has expired.

The invention further provides a method for dynamically controlling access to a bus having a plurality of devices operationally connectable thereto, each of said devices having a priority level associated therewith, said method comprising the step of:

receiving a request to control the bus and a priority level associated with the request from a second of said devices; and

transmitting the priority level associated with the request received from the second device and an associated pending request signal to a first of said devices having control of access to the bus in response to the receipt of the request from the second device.

Preferably the transmitting step is preceded by the steps of: comparing the priority level associated with the second device with the priority levels associated with each device other than the first device to identify the priority level having the highest level; and substituting the identified priority level for the second priority level.

In the preferred embodiment there is an initial step of requesting control of the bus by the first device of the plurality of devices from the bus controller wherein the request by the first device has a priority level associated therewith. This request is received at the bus controller, which arbitrates the priority levels associated with requests received from various devices to identify a first priority level having the greatest value among the received priority levels and for identifying the device associated with the identified first priority level as a first device. Control of the bus can then be granted to the first device in response to the arbitrating step to allow the transfer of data such as applications data, computer program instructions, and address data between the first device and said bus.

In another preferred embodiment, the controlling step comprises the steps of: granting a request from a first of said devices to control the bus; receiving a request to control the bus and a priority level associated with the request from a second of said devices; transmitting the priority level associated with the request received from the second device and an associated pending request signal to the first device in response to the receipt of the request from the second device; and controlling the duration of the control of the bus by the first device based on a combination of a latency timer associated with the first device and a comparison of the priority level of the first device with the priority level associated with the request from the second device.

Also in this preferred embodiment, said controlling step further comprises the steps of: continuing control of the bus by the first device in response to the latency timer associated with the first device having not expired;

extending the duration of control of the bus by the first device in response to the priority level associated with the first device being at least as great as the priority level associated with the second device and the latency timer associated with the first device having expired; and terminating the control of the bus by the first device in response to the priority level associated with the first device being less than the priority level associated with the second device and the latency timer associated with the first device having expired.

The invention further provides a system for dynamically controlling a bus, said system comprising:

a plurality of devices, each of said devices having a priority level and a latency timer associated therewith; and
a bus controller, operationally connectable to each of said devices, for controlling the duration of control of a bus by a first of said devices based on the combination of the latency timer and the priority level associated with the first device, and a priority level associated with a pending request by a second of said devices to control the bus.

In one embodiment, said bus controller comprises: means for granting a request from a first of said devices to control the bus; means for receiving a request to control the bus and a priority level associated with the request from a second of said devices; feedback means for transmitting the priority level associated with the request received from the second device and the first device in response to the receipt of the request from the second device; and means for controlling the duration of the control of the bus by the first device based on a combination of the latency timer associated with the first device and a comparison of the priority level of the first device with the priority level associated with the request from the second device.

The invention still further provides a system for dynamically controlling access to a bus having a plurality of devices operationally connectable thereto, each of said devices having a priority level associated therewith, said system comprising:

means for receiving a request to control the bus and a priority level associated with the request from a second of said devices; and
feedback means for transmitting the priority level associated with the request received from the second device to a first of said devices having control of access to the bus in response to the receipt of the request from the second device.

In one embodiment said feedback means comprises: determining means for comparing the priority level associated with each device other than the first device from which a control request is received to identify the priority level having the greatest value; and transmitting means for transmitting the identified priority level and a pending request signal to the first device.

The invention still further provides a system for dynamically accessing a bus, said system comprising:

a plurality of devices, each of said devices being operationally connectable to the bus and having a priority level associated therewith; and
duration means for controlling the duration of access to the bus by a first of said devices based on the combination of a latency timer and the priority level associated with the first device, and a priority level associated with a pending request by a second of said devices to control the bus.

The invention still further provides a computer program product for dynamically controlling access to a bus having a plurality of devices operationally connectable thereto, each of said devices having a priority level associated therewith, said computer program product comprising:

a computer readable storage medium having computer readable code means embodied in said medium, said computer readable code means comprising:
computer instruction means for controlling the duration of control of the bus by a first of said devices based on the combination of a latency timer and the priority level associated with the first device, and a priority level associated with a pending request by a second of said devices to control the bus.

The invention still further provides a computer program product for dynamically controlling access to a bus having a plurality of devices operationally connectable thereto, each of said devices having a priority level associated therewith, said computer program product comprising:

a computer readable storage medium having computer readable code means embodied in said medium, said computer readable code means comprising:
computer instruction means for receiving a request to control the bus and a priority level associated with the request from a second of said devices; and
computer instruction means for transmitting the priority level associated with the request received from the second device and an associated pending request signal to a first of said devices having control of access to the bus in response to the receipt of the request from the second device.

It will be appreciated that such systems and computer programs will benefit from the same pr ferred features as the methods of the invention.

Viewed from another aspect the invention provides

a method for dynamically accessing a bus having a plurality of devices operationally connectable thereto, each of said devices having a priority level associated therewith, said method comprising the step of:

controlling the duration of access to the bus by a first of said devices based on the combination of a latency timer and the priority level associated with the first device, and a priority level associated with a pending request by a second of said devices to control the bus.

Preferably said controlling step comprises the steps of: receiving control of the bus by a first device from a bus controller in response to a request by the first device and a priority level associated therewith; and receiving a second priority level from the bus controller, said second priority level associated with a pending control request received by the bus controller from a second device. It is also preferred that said controlling step further comprises the step of: controlling the duration of the control of the bus by the first device based on a combination of a latency timer associated with the first device and a comparison of the priority level of the first device with the second priority level associated with the request from the second device. It is additionally preferred that said receiving control of the bus by a first device step is preceded by the step of: requesting control of the bus by the first device of the plurality of devices from the bus controller wherein the request by the first device has a priority level associated therewith.

Thus the systems, methods and computer program products described herein seek to improve bus performance in a computer system in which multiple devices share a common bus, and in particular to increase throughput and decrease latency, whilst accommodating dynamically changing bus usage conditions. The access of multiple devices to a common bus is therefore controlled in the preferred embodiment by combining the effect of priority levels and latency, and by dynamically controlling the latency timer of a device.

As described in more detail below, in the preferred embodiment, each master in the system has an associated latency timer and a priority level indicator. The latency timer may be implemented using a register representing the maximum number of cycles that the device may control the bus and a counter for counting the cycles of control. The priority level indicator represents the priority level associated with the device. The system dynamically controls the bus by using a bus controller which is operationally connected to each of the devices. The bus controller controls the duration that a first device has control over the bus based on the combination of the latency timer associated with the first device, the priority level associated with the first device and the priority level associated with a pending request received by the bus controller from a second device. In addition, the system prevents all other devices from controlling the bus while one device has control of the bus. The bus controller comprises a means for granting a request from a first device to control the bus. The bus controller also includes means for receiving a request to control the bus and an associated priority level from a second device, and feedback means for transmitting the priority level associated with the request received from the second device to the first device in response to the receipt of the request from the second device. The bus controller also includes means for controlling the duration of the control of the bus by the first device based on a combination of the latency timer associated with the first device and the results of a comparison of the priority level of the first device with the priority level associated with the request from the second device. Still further, the bus controller may also comprise a means for continuing control of the bus by the first device if the latency timer of the first device has not expired, even if the priority level of the second device is greater than the priority level of the first device. An extension means is also provided for extending the duration of control of the bus by the first device if the priority level associated with the first device is greater than or equal to the priority level associated with the second device, even if the latency timer associated with the first device has expired. The system will terminate or relinquish control of the bus by the first device if the priority level associated with the first device is less than the priority level associated with the second device and the latency timer associated with the first device has expired.

In a first alternative embodiment, each of the devices has a priority level associated with it, and a first device has control of the bus. The system receives a request to control the bus and an associated priority level from a second device, and compares the priority level associated with each device other than the first device from which a control request is received to identify the priority level having the greatest value. The identified priority level having the greatest value and a corresponding pending request signal are fed back to the first device.

In a second alternative embodiment, each device also has a priority level associated with it, and the system controls the duration of access to the bus by the first device based on the combination of the latency timer and the priority level associated with the first device, and a priority level associated with a pending request by a second device to control the bus. In addition, the first device receives control of the bus from a bus controller in response to its request to control the device and the priority level associated with it. The first device also receives a second priority level from the bus controller which is associated with a pending control request received by the bus controller from a second device. The first device continues to control the bus if its latency timer has not expired, even if its priority level is less than the second priority level associated with the request from a second device. The first device extends the duration of its control of the bus in response to its associated priority level being greater than or equal to the priority level associated with a second device, even if the

latency timer associated with the first device has expired. Finally, the first device relinquishes or terminates its control of the bus if its associated priority level is less than the priority level associated with the second device and the latency timer associated with the first device has expired.

The approach described herein combines the use of priority and latency to improve bus performance, wherein the latency timer of the device having control of the bus may be dynamically changed. This is accomplished by overriding an expired latency timer and allowing the corresponding device to continue control of the bus as long as its priority is at least as great as the pending request priority indicator. Once the pending request priority indicator is greater than the priority level of the device presently controlling the bus, the device presently controlling the bus will relinquish or terminate control of the bus.

A preferred embodiment of the invention will now be described in detail by way of example only with reference to the following drawings:

FIG. 1 is a block diagram illustrating a computer system;

FIG. 2 is a block diagram illustrating the relationship between the master devices, bus controller, and a slave device;

FIG. 3 is a block diagram illustrating the feedback logic circuit of the bus controller illustrated in FIG. 2;

FIG. 4 is a flow chart representation illustrating the operation of the dynamically controlling system; and

FIG. 5 is a timing diagram illustrating exemplary operations of the dynamic controlling system.

Referring to Figure 1, a block diagram shows a computer system illustrated generally at 100, computer system 100 preferably being on a single integrated circuit chip. Computer system 100 has a number of devices and a bus. In particular, computer system 100 has several master devices including a 4XX PowerPC ("PPC") central processing unit ("CPU") 102, a direct memory access ("DMA") controller 104 and a PowerPC local bus ("PLB") master 106. In addition, computer system 100 also has an external bus interface unit ("EBIU") 108, which includes a dynamic random access memory ("DRAM") controller 108a and an input/output ("I/O") controller 108b. Still further, computer system 100 also has an on-chip peripheral bus ("OPB") bridge 110, an OPB master 112, an OPB slave 114, a parallel port 116 and a serial port 118.

Finally, the computer system 100 also has two bus controllers, a bus controller referred to generally at 120 and a bus controller referred to generally at 122. Bus controller 120 includes arbiter 130, feedback logic circuit 131 and PowerPC 4XX Local Bus 132. Bus controller 122 includes arbiter 134, feedback logic circuit 135 and on-chip peripheral bus 136.

4XX PPC CPU 102, DMA controller 104 and PLB master 106 are examples of "master" devices, and may be referred to as "Master 1," "Master 2," and "Master 3," respectively. EBIU 108 and OPB bridge 110 are examples of "slave" devices.

Master 1 102, Master 2 104, and Master 3 106 are each operationally connectable to bus controller 120. In addition, slave 108 and slave 110 are also operationally connectable to bus controller 120. In the computer system illustrated generally at 100, slave 110 (i.e., OPB bridge), acts as a slave device and operationally connects one of Master 1 102, Master 2 104, and Master 3 106 to bus controller 122.

As illustrated in Figure 1, the approach described herein may facilitate the interconnection of a plurality of master devices or functions to a commonly shared bus, but similarly, may also enable the interconnection of a plurality of slave devices to a commonly shared bus. In a preferred embodiment, the master functions and the slave functions are contained within a single integrated chip, but the same approach may be used to connect devices located on a single chip such as computer system 100 to devices located on another chip via external bus 140.

Now referring to Figure 2, the operation of the dynamic controlling system, including bus controller 120 will be described. Each master device has a latency timer. For example, Master 1 102 has latency timer 102a, Master 2 104 has latency timer 104a, and Master 3 106 has latency timer 106a.

In one embodiment, the bus arbitration scheme implemented by bus controller 120 uses a dynamic priority scheme. Under this scheme, each master has an associated priority level. In one particular priority scheme, the priority level may be one of four levels represented by a two- bit request priority signal. The request priority levels are, in order from highest priority to lowest priority, as follows: "11" representing a priority level of "high", "10" representing a priority level of "medium high", "01" representing a priority level of "medium low", and "00" representing a priority level of "low". It will be understood by those having skill in the art that a variety of different priority schemes may be used.

In operation, a master submits a bus request and a corresponding request priority to the bus controller 120. For example, Master 1 102, when submitting a bus request, will simultaneously submit a bus request signal 202 and a request priority signal 204 to bus controller 120.

If bus controller 120 receives bus requests simultaneously from more than one master, arbiter 130 determines if bus 132 is available, compares all of the request priority levels and grants control of bus 132 to the master having the highest priority level. Once the arbiter 130 has granted control of bus 132 to the master which submitted the highest request priority in conjunction with its bus request, feedback logic circuit 131 then determines the level of the highest request priority among those de-

vices having bus requests which are still pending. The resulting current pending request priority, together with the fact that another request is pending, is then fed back to each master across links 151 and 152, respectively. This information is then used by the master which currently has control of the bus to determine whether or not to terminate its control of the bus before it has completed its data transfer in order to let another master use the bus. In other words, the master currently having control of the bus may continue using the bus until all data which it currently needs to transfer has been transferred by dynamically changing its latency timer (i.e., overriding its expired latency timer) depending on the currently pending request priority.

Referring to Figure 2, Master 1 102 submits its bus request 202 and request priority 204 to bus controller 120. Similarly, Master 2 submits its bus request 212 and request priority 214 to bus controller 120. Finally, Master 3 106 submits its bus request 222 and request priority 224 to bus controller 120. Once bus controller 120 receives one or more bus requests and corresponding request priorities, arbiter 130, which is a component of bus controller 120, identifies the master which submitted a bus request with the highest request priority, and grants bus 132 to the corresponding master. The master which was granted the bus receives a grant signal, turns off its request signal, and begins sending a burst transfer. By way of example, assuming arbiter 130 determined that Master 1 102 had the highest priority, arbiter 130 sends a grant 208 to Master 1 102, Master 1 102 receives the grant 208, turns off its request 202 and begins sending "burst" data 206 to bus controller 120 which, via bus 132, sends the burst to slave device 108. Similarly, if Master 2 104 had the highest priority, it receives the grant 218, turns off its request 212, and begins sending "burst" data 216. Finally, if Master 3 106 had the highest priority, it receives the grant 228, turns off its request 222 and begins sending "burst" data 226.

Feedback logic circuit 131 processes the remaining bus requests during each clock cycle. In particular, feedback logic circuit 131 compares the bus requests signals to determine if any requests are still pending. In addition, feedback logic circuit 131 also compares the request priority levels for each pending request to identify the highest pending request priority among the currently pending requests. Circuit 131 then transmits the value of the highest current pending request priority and the fact that a request is pending back to each master device on links 151 and 152, respectively, during each clock cycle. It will be understood by those skilled in the art that the pending request priority signal and pending request signal may be fed back to the master devices separately or as one combined signal.

Referring to Figure 3, a preferred embodiment of feedback logic circuit 131 is illustrated. As illustrated in Figure 3, the bus requests of the masters, namely, Master 1 request 202, Master 2 request 212, and Master 3 request 222 are compared by feedback logic circuit 131 using a logic "OR" gate 240. If any of the masters currently have a bus request pending, the output of OR gate 240 will be high or a logic "1". The output of logic "OR" gate 240 represents the request pending signal which is fed back by link 152 to each master.

In addition, the bus request signals from each of the masters are also combined with their corresponding request priority level using "AND" gate logic. Although only a single communications line (e.g. the communications line between Master 1 request priority 204 and "AND" gate 242) is shown, it will be understood by those skilled in the art that this one line represents collectively the total input lines for the request priority signal to the "AND" gate 242. In one particular embodiment, Master 1 bus request 202 is combined with its corresponding Master 1 request priority 204 using logic "AND" gate 242. Master 2 request 212 and Master 2 request priority 214 are combined using logic "AND" gate 244. Finally, Master 3 request 222 and Master 3 request priority 224 are combined using logic "AND" gate 246. The results of the comparisons of logic "AND" gates 242, 244 and 246 are compared using comparator 248. The result of comparator 248 represents the pending request priority signal which indicates the highest priority level among all currently pending bus requests. The pending request priority signal is fed back to each master across link 151.

Figure 4 is a flow chart and Figure 5 is a timing diagram illustrating the above procedure. Referring first to Figure 4, a flow chart of the operations of the dynamic controlling system is shown. The operations illustrated in the flow chart of Figure 4 are viewed from the context of a master device. The system begins when a master makes a bus request at 402. The master then begins checking at 404 during each clock cycle to determine if it has received a bus grant from the arbiter. If no grant is received, the master continues to request the bus and check its grant signal during each clock cycle. If a grant signal is received by the master at 404, the master takes control of the bus, sets its latency counter to "0", and may start a burst transfer at 406.

Thereafter, during each cycle, the master checks to see if its latency counter has expired at 408. If its latency counter has not expired, the master then determines if it still needs to control the bus (e.g., if it has more transfers to send across the bus) at 412. If the master has more transfers to send across the bus, the master then continues its burst at 414 and repeats the operations of decision blocks 408, 410 and 412 during each cycle. If the master determines at 412 that it no longer needs the bus (e.g., it has no more transfers to make across the bus), the master releases the bus at 416 which completes the data transfer.

If the master determines at 408 that its latency counter has expired, it then determines at 424 as to whether the pending request priority is greater than its own priority. If the pending request priority is greater than the master's priority, the master releases the bus at 430. If the pending request priority is not greater than

its own priority, the master then determines whether it has more transfers to make at 426. If it has more transfers to make (or needs to continue controlling the bus), the master continues the burst during the present cycle at 428 and returns to 424 to continue the comparison process of the pending request priority versus its own priority during the next cycle. If the master determines at 426 that it no longer needs the bus, the master then releases the bus at 427 and stops.

Once the master releases the bus at 430, it determines at 432 whether it has more transfers to be made (i.e., it needs control of the bus again). If the master determines at 432 that it does not need to control the bus for additional cycles, it stops processing until it submits a new bus request to the bus controller. On the other hand, if the master determines at 432 that it needs control of the bus again for additional cycles, the master then submits another request to the bus controller at 402. The processing then continues as described above.

Referring to Figure 5, a timing diagram illustrating an example of the operation is shown. In the particular example illustrated in Figure 5, arbiter 130 receives a bus request from Master 1 during Cycle 1 having a request priority of "10." Since no other requests are pending, arbiter 130 sends a grant signal back to Master 1 during Cycle 2. In response to receiving a grant signal, Master 1 turns off its request, resets its latency counter and begins controlling the bus during Cycle 2. In the example illustrated in Figure 5, Master 1 immediately begins sending a burst across the bus during Cycle 2. Also during Cycle 2, feedback logic circuit 131 (see Figure 2) determines that no other request is pending and feeds a "0" back to each master.

During Cycle 3, Master 2 submits a bus request having a request priority of "10." Feedback logic circuit 131 determines that Master 2's bus request is pending, and that it has a request priority of "10." As a result, feedback logic circuit 131 sends signals back to each master during Cycle 3 indicating that a request is pending and that the pending request priority is "10." During Cycle 3, Master 1 receives the pending request priority signal and the request pending signal from feedback logic circuit 131. In response, Master 1 compares the pending request priority of "10" against its own priority of "10," checks its latency timer to see if it has expired, and determines that it can continue to control the bus. As such, Master 1 continues to control the bus during Cycle 3.

Bus controller 120 receives a bus request from Master 3 during Cycle 4. This bus request has a priority level of "01." In response, feedback logic circuit 131 determines that bus requests are currently pending and also compares the request priority levels and determines that the highest pending request priority is "10." As a result, feedback logic circuit 131 sets the request pending signal to "1" and also sets the pending priority signal to "10" corresponding to the highest pending request priority level, and transmits these signals back to each master.

In response to the receipt of the request pending signal and the pending request priority, Master 1 compares the pending request priority to its own priority level during Cycle 4. Master 1 determines that its own priority is at least as great as the pending request priority during Cycle 4. However, also during Cycle 4, Master 1 determines that its latency timer has expired. Nonetheless, since its priority is at least as great as that of the pending request priority, Master 1 dynamically changes its latency timer by overriding it, and continues to control the bus and continues its "burst." This process continues during Cycles 5 and 6.

During Cycle 7, Master 3 decides to submit a different bus request having a request priority of "11." In response, feedback logic circuit 131 determines that a request is pending, and that the highest request pending priority level is "11." As a result, feedback logic circuit 131 sends a request pending signal and a request pending priority level of "11" back to each master during Cycle 7.

Master 1 receives the request pending signal and the request pending priority level of "11" during Cycle 7 from feedback logic circuit 131. In response, Master 1 compares the request pending priority level of "11" to its own priority level of "10" and determines that the request pending priority level is greater. As a result, Master 1 stops its burst and relinquishes control of the bus control during Cycle 7.

During Cycle 8, arbiter 130 determines that the bus is available, and based on a comparison of the priority levels of the pending requests for Master 2 and Master 3, determines that Master 3 has the highest pending request priority, and grants the bus to Master 3 during Cycle 8. As a result, Master 3 begins controlling the bus during Cycle 8 and initiates a burst on the bus. Feedback logic circuit continues processing as described above during Cycles 9 and 10. Based on Master 3's priority and latency timer, Master 3 continues its "burst" across the bus until it finishes its burst during Cycle 10. Master 3 then relinquishes the bus during Cycle 10 upon completion of its burst.

Finally, arbiter 130 determines that the bus is available during Cycle 11. As a result, arbiter 130 determines that Master 2's bus request is still pending and that it has the greatest pending request priority. Arbiter 130 grants the bus to Master 2 during Cycle 11, and, in response, Master 2 begins controlling the bus during Cycle 11 and begins a burst across the bus.

Note that the arbiter 130 may utilize any number of known methods of selecting competing priority levels. Implementation of arbiter 130, therefore, may be dependent upon the desired arbitration scheme.

It will be understood that the invention is not restricted to the transfer of any specific type of data or signals on the bus, but applies to the transfer of any type of data such as instruction data, address data, data bus and transfer qualifiers, burst signals, bus lock signals, and

so forth. Further, whilst the preferred embodiment has predefined slave and master status, it will be appreciated by those of skill in the art that any particular device may be a slave device for one operation and a master device for a different operation. It will also be recognised that the present invention applies not only to "burst" transfers, but may also apply to bus locking mechanisms and other uses relating to bus control which may affect bus performance. It will further be understood that rather than using positive logic such that a positive voltage corresponds to a logic state of "1", any suitable means of signifying an active and inactive state may be utilised.

It will also be appreciated by those skilled in the art that the present invention may take the form of an entirely hardware embodiment, an entirely software embodiment, or an embodiment combining software and hardware aspects. Typically each block or step of the timing diagram and flow chart is implemented by computer program instructions loaded onto a computer or other programmable data processing apparatus to produce a machine such that the instructions which execute on the computer or other programmable apparatus create means for implementing the functions specified in the timing diagram or flow chart block(s) or step(s). These computer program instructions are generally stored in a computer readable memory that can direct a computer or other programmable apparatus to function in a particular manner by causing a series of operational steps to be performed on the computer or other programmable apparatus.

Claims

1. A method for dynamically controlling access to a bus having a plurality of devices operationally connectable thereto, each of said devices having a priority level associated therewith, said method comprising the step of:

   controlling the duration of control of the bus by a first of said devices based on the combination of a latency timer and the priority level associated with the first device, and a priority level associated with a pending request by a second of said devices to control the bus.

2. The method of Claim 1 wherein said controlling step includes the steps of:

   granting by a bus controller a request from the first device to control the bus;
   receiving at the bus controller a request to control the bus and a priority level associated with the request from the second device; and
   transmitting from the bus controller the priority level associated with the request received from the second device and an associated pending request signal to the first device in response to the receipt of the request from the second device.

3. The method of Claim 2 wherein said controlling step further includes the steps of:

   receiving control of the bus by the first device from the bus controller in response to said granted request; and
   receiving at the first device the priority level from the bus controller associated with the pending control request from the second device.

4. The method of Claim 3 wherein said controlling step further includes the step of:

   comparing the priority level of the first device with the priority level associated with the request from the second device.

5. The method of Claim 4 wherein said controlling step further includes the steps of:

   continuing control of the bus by the first device until the latency timer associated with the first device has expired;
   extending the duration of control of the bus by the first device in response to the priority level associated with the first device being at least as great as the priority level associated with the second device after the latency timer associated with the first device has expired; and
   terminating the control of the bus by the first device in response to the priority level associated with the first device being less than the priority level associated with the second device after the latency timer associated with the first device has expired.

6. A method for dynamically controlling access to a bus having a plurality of devices operationally connectable thereto, each of said devices having a priority level associated therewith, said method comprising the step of:

   receiving a request to control the bus and a priority level associated with the request from a second of said devices; and
   transmitting the priority level associated with the request received from the second device and an associated pending request signal to a first of said devices having control of access to the bus in response to the receipt of the request from the second device.

7. The method of Claim 6 wherein the transmitting step is preceded by the steps of:

comparing the priority level associated with the second device with the priority levels associated with each device other than the first device to identify the priority level having the highest level; and
substituting the identified priority level for the second priority level.

8. The method of any preceding Claim further comprising the initial step of requesting control of the bus by the first device of the plurality of devices from the bus controller wherein the request by the first device has a priority level associated therewith.

9. The method of Claim 8 wherein said initial step further comprises the steps of:

receiving at the bus controller at least one request to control the bus and a priority level associated with the request from the plurality of devices;
arbitrating the priority levels associated with the requests received from the devices to identify a first priority level having the greatest value among the received priority levels and for identifying the device associated with the identified first priority level as a first device; and
granting control of the bus to the first device in response to the arbitrating step.

10. The method of any preceding Claim further comprising the step of: transferring data between the first device and said bus in response to said controlling step.

11. The method of Claim 10 wherein said data comprises at least one of applications data, computer program instructions, and address data.

12. A system for dynamically controlling a bus, said system comprising:

a plurality of devices, each of said devices having a priority level and a latency timer associated therewith; and
a bus controller, operationally connectable to each of said devices, for controlling the duration of control of a bus by a first of said devices based on the combination of the latency timer and the priority level associated with the first device, and a priority level associated with a pending request by a second of said devices to control the bus.

13. A system for dynamically controlling access to a bus having a plurality of devices operationally connectable thereto, each of said devices having a priority level associated therewith, said system comprising:

means for receiving a request to control the bus and a priority level associated with the request from a second of said devices; and
feedback means for transmitting the priority level associated with the request received from the second device to a first of said devices having control of access to the bus in response to the receipt of the request from the second device.

14. A system for dynamically accessing a bus, said system comprising:

a plurality of devices, each of said devices being operationally connectable to the bus and having a priority level associated therewith; and
duration means for controlling the duration of access to the bus by a first of said devices based on the combination of a latency timer and the priority level associated with the first device, and a priority level associated with a pending request by a second of said devices to control the bus.

15. A computer program product for dynamically controlling access to a bus having a plurality of devices operationally connectable thereto, each of said devices having a priority level associated therewith, said computer program product comprising:

a computer readable storage medium having computer readable code means embodied in said medium, said computer readable code means comprising:
computer instruction means for controlling the duration of control of the bus by a first of said devices based on the combination of a latency timer and the priority level associated with the first device, and a priority level associated with a pending request by a second of said devices to control the bus.

16. A computer program product for dynamically controlling access to a bus having a plurality of devices operationally connectable thereto, each of said devices having a priority level associated therewith, said computer program product comprising:

a computer readable storage medium having computer readable code means embodied in said medium, said computer readable code means comprising:
computer instruction means for receiving a request to control the bus and a priority level associated with the request from a second of said devices; and
computer instruction means for transmitting the priority level associated with the request received from the second device and an associ-

ated pending request signal to a first of said devices having control of access to the bus in response to the receipt of the request from the second device.

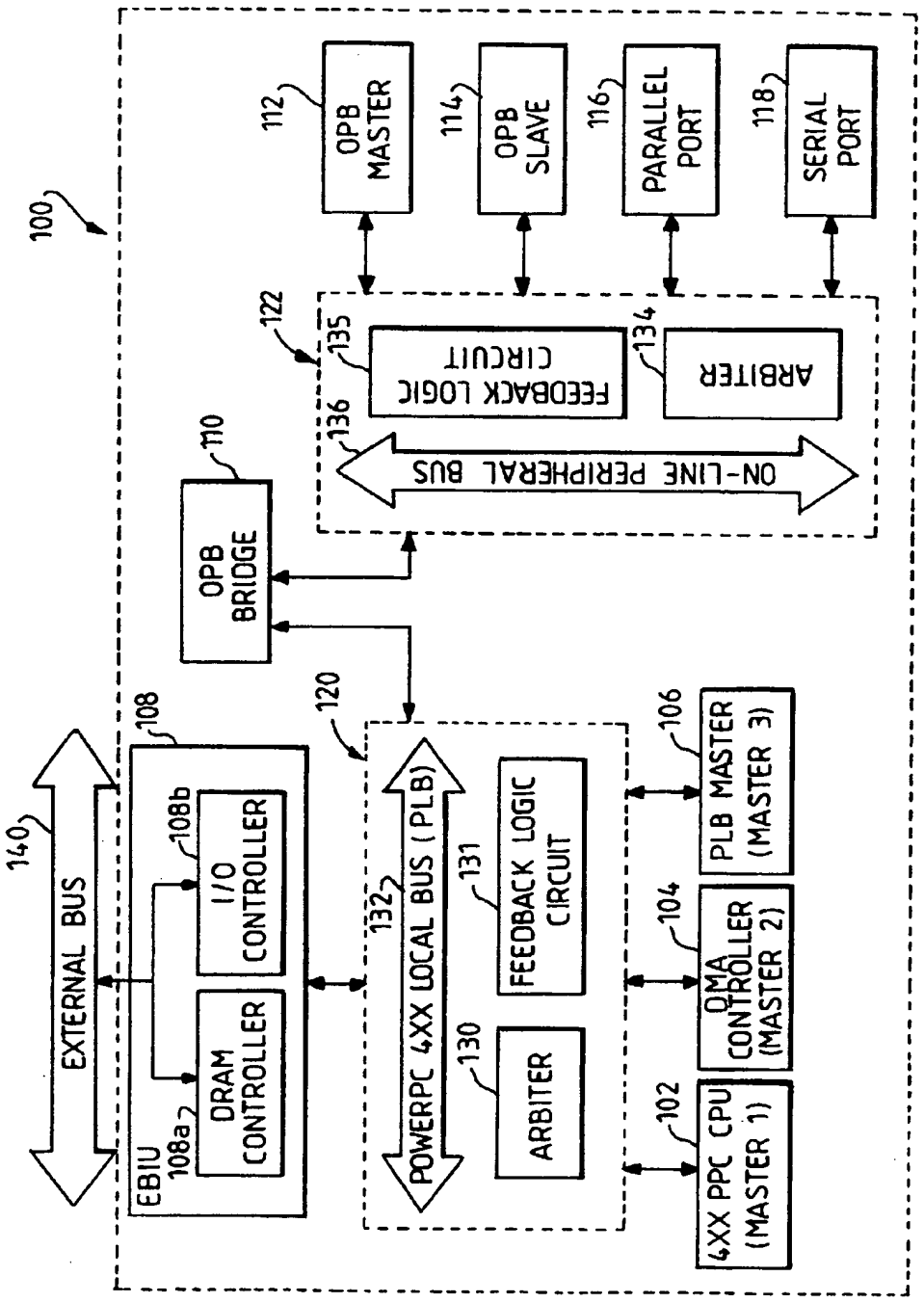
100
112
OPB MASTER
114
OPB SLAVE
116
PARALLEL PORT
118
SERIAL PORT
122
135
FEEDBACK LOGIC CIRCUIT
134
ARBITER
136
ON-LINE PERIPHERAL BUS
110
OPB BRIDGE
120
108
140
EXTERNAL BUS
108b
I/O CONTROLLER
DRAM CONTROLLER
EBIU
108a
132
POWERPC 4XX LOCAL BUS (PLB)
131
FEEDBACK LOGIC CIRCUIT
130
ARBITER
106
PLB MASTER (MASTER 3)
104
DMA CONTROLLER (MASTER 2)
102
4XX PPC CPU (MASTER 1)

FIG. 1

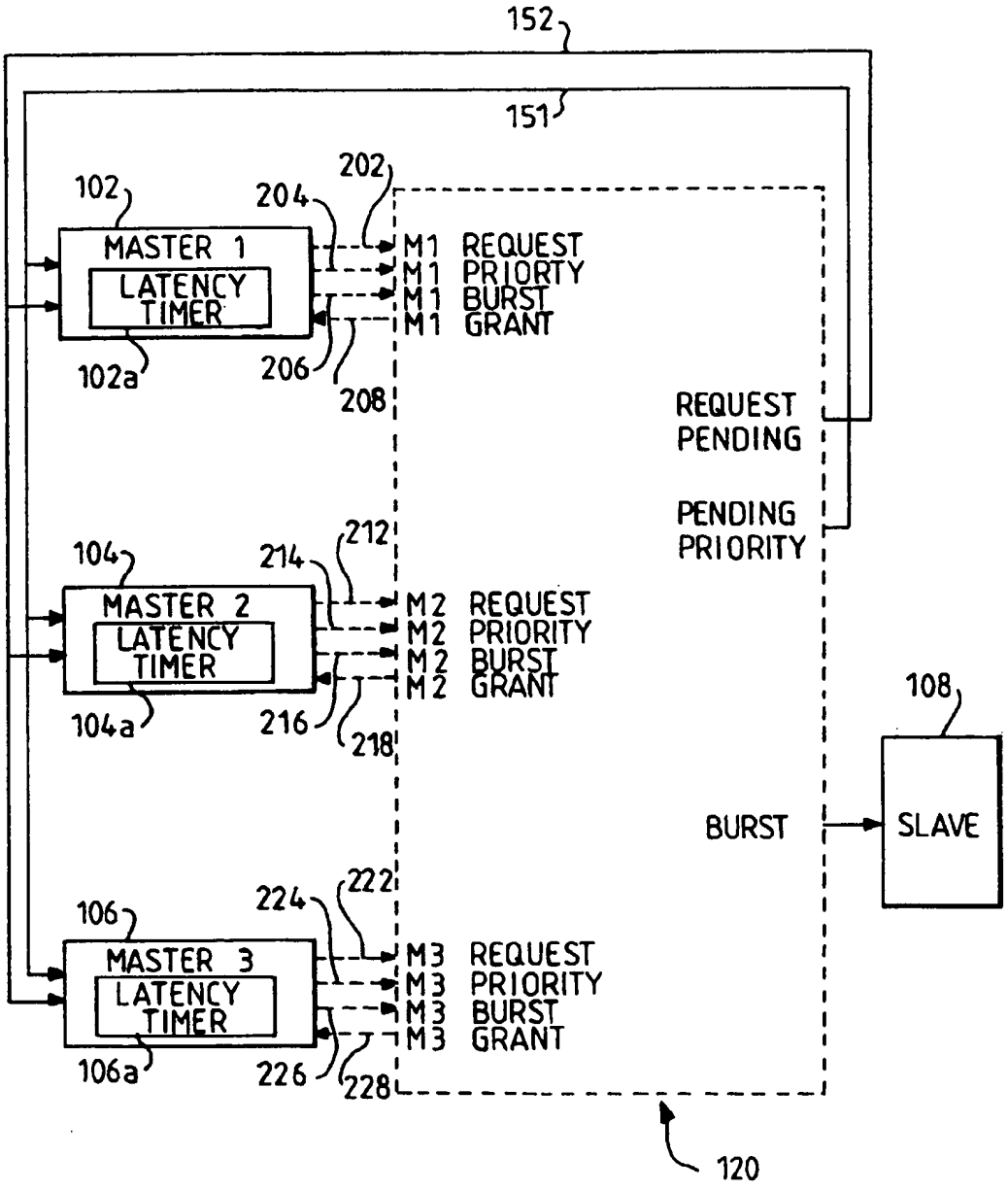
152
151
202
102
204
MASTER 1
LATENCY TIMER
M1 REQUEST
M1 PRIORTY
M1 BURST
M1 GRANT
102a
206
208
REQUEST PENDING
PENDING PRIORITY
212
104
214
MASTER 2
LATENCY TIMER
M2 REQUEST
M2 PRIORITY
M2 BURST
M2 GRANT
104a
216
218
108
BURST
SLAVE
222
224
106
MASTER 3
LATENCY TIMER
M3 REQUEST
M3 PRIORITY
M3 BURST
M3 GRANT
106a
226
228
120

FIG. 2

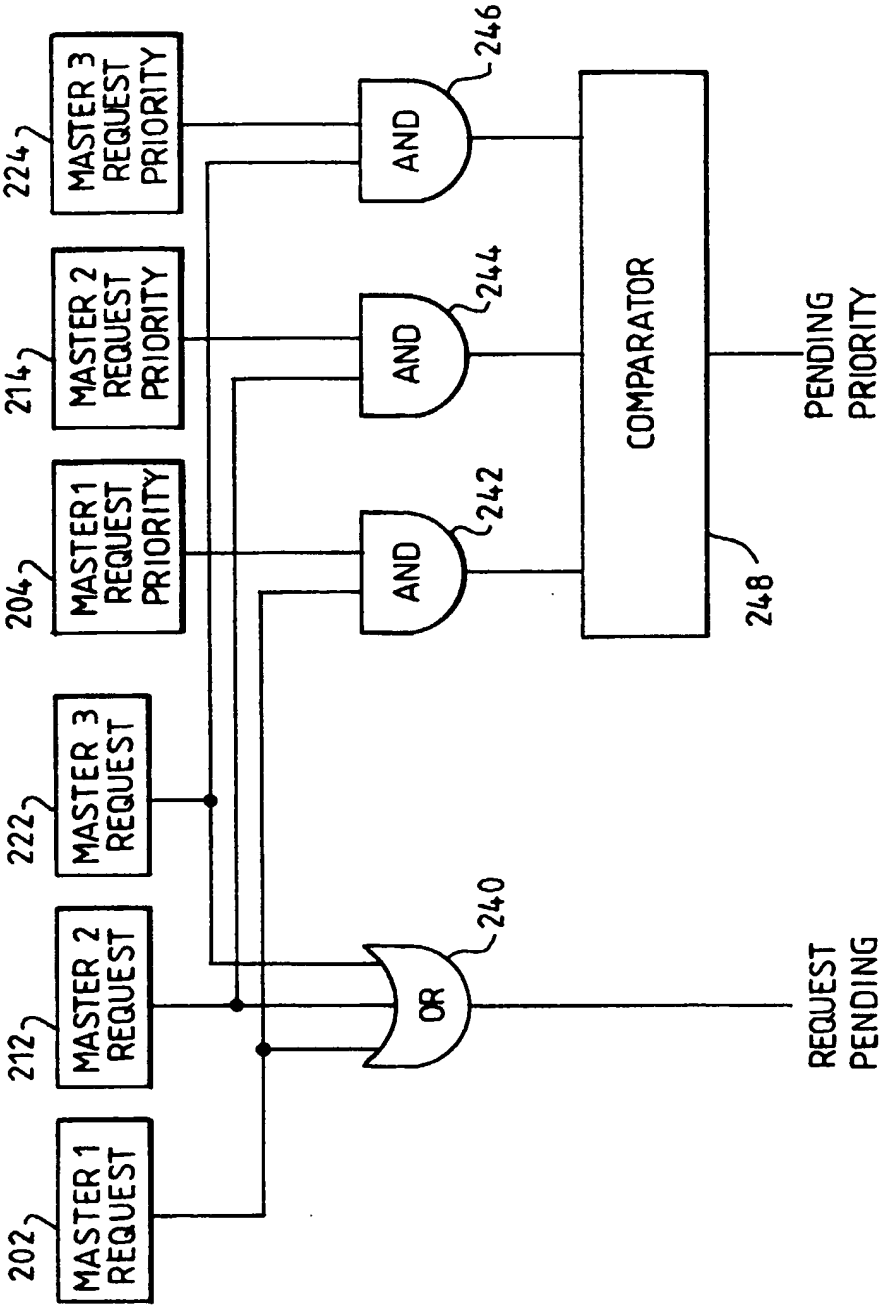
202
MASTER 1 REQUEST
212
MASTER 2 REQUEST
222
MASTER 3 REQUEST
204
MASTER1 REQUEST PRIORITY
214
MASTER 2 REQUEST PRIORITY
224
MASTER 3 REQUEST PRIORITY
OR
240
AND
242
AND
244
AND
246
COMPARATOR
248
REQUEST PENDING
PENDING PRIORITY

FIG. 3

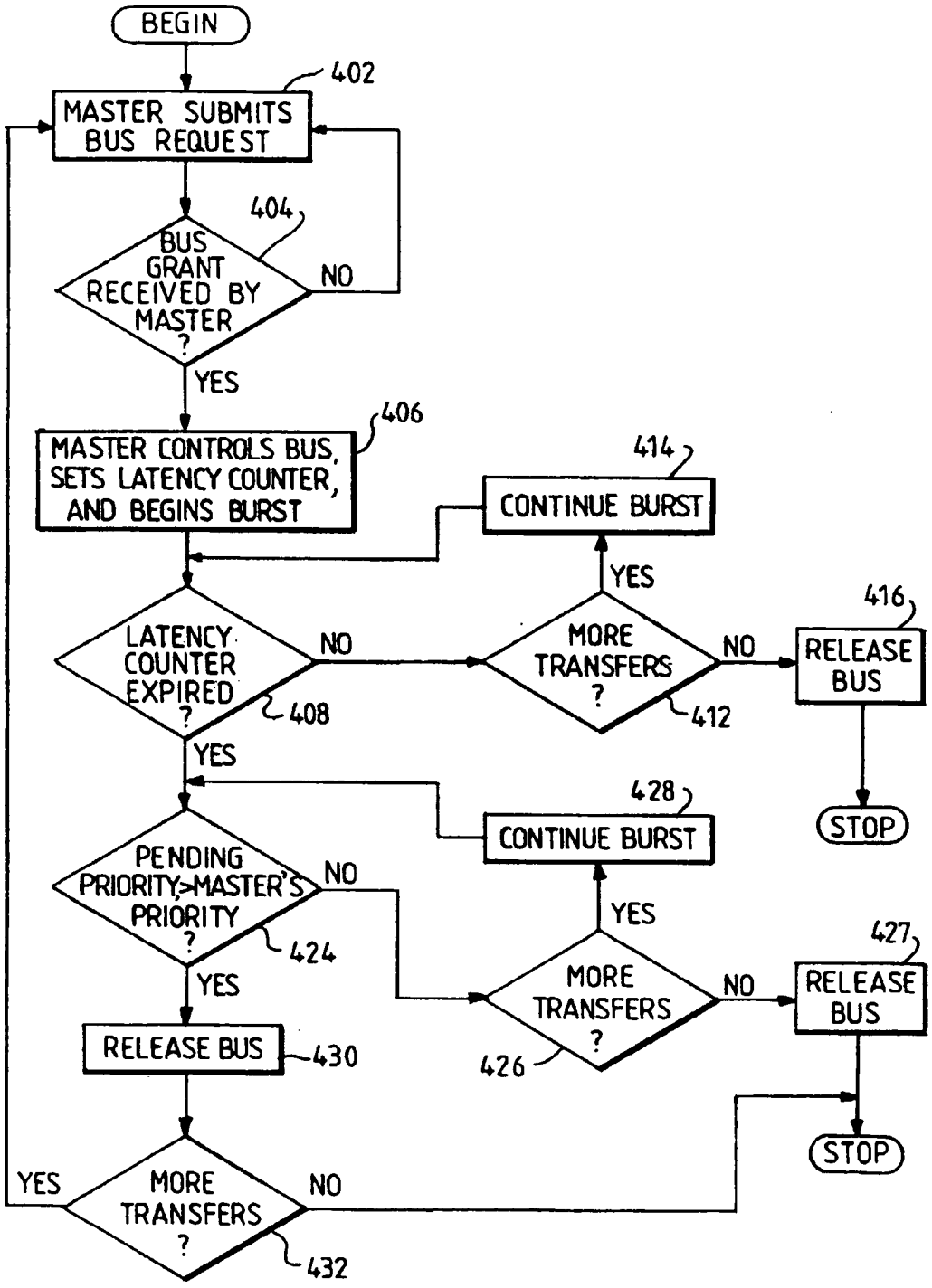
BEGIN
402
MASTER SUBMITS BUS REQUEST
404
BUS GRANT RECEIVED BY MASTER ?
NO
YES
406
MASTER CONTROLS BUS, SETS LATENCY COUNTER, AND BEGINS BURST
414
CONTINUE BURST
YES
LATENCY COUNTER EXPIRED ?
NO
408
MORE TRANSFERS ?
NO
412
416
RELEASE BUS
STOP
YES
428
CONTINUE BURST
PENDING PRIORITY>MASTER'S PRIORITY ?
NO
424
YES
YES
MORE TRANSFERS ?
NO
426
427
RELEASE BUS
STOP
RELEASE BUS
430
YES
MORE TRANSFERS ?
NO
432

FIG. 4

CYCLE	1	2	3	4	5	6	7	8	9	10	11	12
CLOCK												
BUS												
M1 REQUEST	1											
M1 PRIORITY	10											
M1 GRANT		1										
M1 LATENCY		RESET		EXPIRED	EXPIRED	EXPIRED	EXPIRED					
M1 BURST												
M2 REQUEST			1	1	1	1	1	1	1	1		
M2 PRIORITY			10	10	10	10	10	10	10	10		
M2 GRANT											1	
M2 LATENCY											RESET	
M2 BURST												
M3 REQUEST				1	1	1	1					
M3 PRIORITY				01	01	01	11					
M3 GRANT								1				
M3 LATENCY								RESET				
M3 BURST												
REQUEST PENDING	1	0	1	1	1	1	1	1	1	1	0	0
PENDING REQUEST PRIORITY	10	0	10	10	10	10	11	10	10	10	0	0

FIG. 5